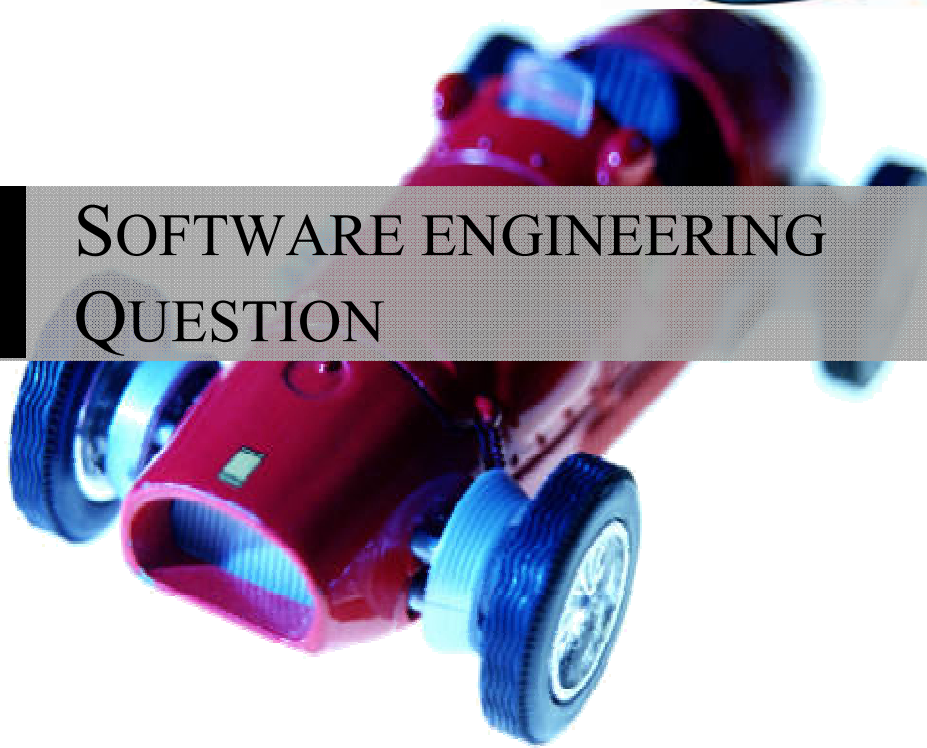




BCA  
NOTES

SOFTWARE ENGINEERING  
QUESTION





# INDEX

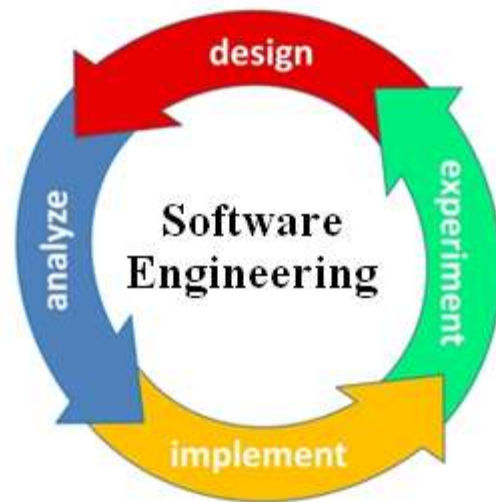
S.No	Question	Page No.
1	What is software engineering?	
2	What is system design?	
3	What is a process frame work?	
4	What are generic framework activities?	
5	What is the role of SRS in requirement analysis?	
6	Define Risk Management?	
7	Write reason for failure of waterfall model?	
8	What is the objective of project planning?	
9	What is DFD?	
10	What is prototype model?	
11	What is UML?	
12	What is estimation?	
13	How many software testing are there .Explain black box, white box, unit, integration, system testing's?	
14	Define software metrics and software measurement?	
15	Explain SDLC with diagram?	
16	Explain software reuse and reengineering?	
17	What is SQA?	

## Software Engineering Must Read QUESTIONS

### TWO MARKS TYPE QUESTIONS

#### 1. What is software engineering?

Ans:-**Software engineering** is the study and application of engineering to the design, development, and maintenance of software. Software engineering (SE) is concerned with developing and maintaining software systems that behave reliably and efficiently, are affordable to develop and maintain, and satisfy all the requirements that customers have defined for them. It is important because of the impact of large, expensive software systems and the role of software in safety-critical applications. It integrates significant mathematics, computer science and practices whose origins are in engineering.

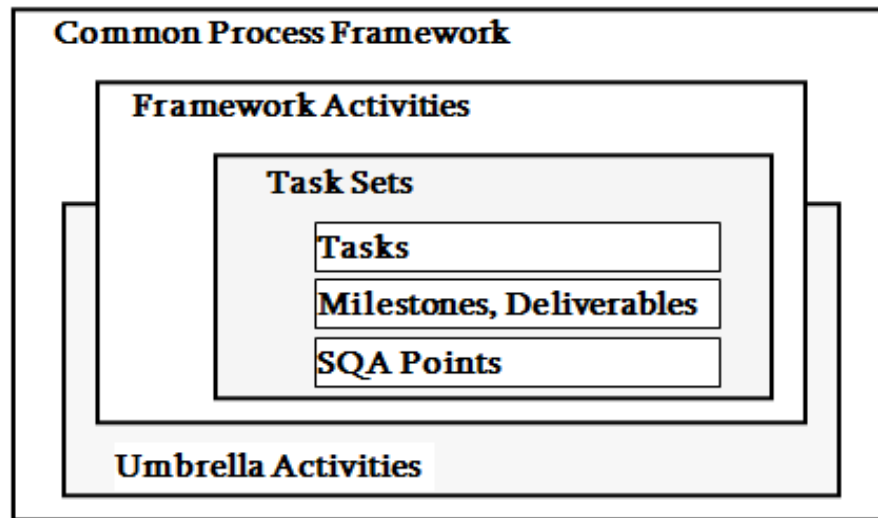


#### 2. What is system design?

Ans: -**Systems design** is the process of defining the architecture, components, modules, interfaces, and data for a **system** to satisfy specified requirements. **Systems design** could be seen as the application of **systems** theory to product development.

#### 3. What is a process frame work?

Ans: - **Software Process Framework:** A process framework establishes the foundation for a complete software process by identifying a small number of framework activities that are applicable to all software projects, regardless of size or complexity. It also includes a set of umbrella activities that are applicable across the entire software process. Some most applicable framework activities are described below.



**Figure: Chart of Process Framework**

**Communication:** This activity involves heavy communication with customers and other stakeholders in order to gather requirements and other related activities.

**Planning:** Here a plan to be followed will be created which will describe the technical tasks to be conducted, risks, required resources, work schedule etc.

**Modeling:** A model will be created to better understand the requirements and design to achieve these requirements.

**Construction:** Here the code will be generated and tested.

**Deployment:** Here, a complete or partially complete version of the software is represented to the customers to evaluate and they give feedbacks based on the evaluation.

#### **4. What are generic framework activities?**

**Ans:-**

#### **5. What is the role of SRS in requirement analysis?**

**Ans:** - Software requirements specification establishes the basis for agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules

#### **6. Define risk management?**

Ans:-The term risk is defined as the potential future harm that may arise due to some present actions. . Risk management in software engineering is related to the various future harms that could be possible on the software due to some minor or non-noticeable mistakes in software development project or process. “Software projects have a high probability of failure so effective software development means dealing with risks adequately .Following are the major steps involved in risk management:-

- Risk index
- Risk analysis
- Risk assessment
- Risk classification.

**Definition:** In the world of finance, risk management refers to the practice of identifying potential risks in advance, analyzing them and taking precautionary steps to reduce/curb the risk.

## 7. Write reason for failure of waterfall model?

**Ans: -Reason for failure of waterfall model:**

- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.

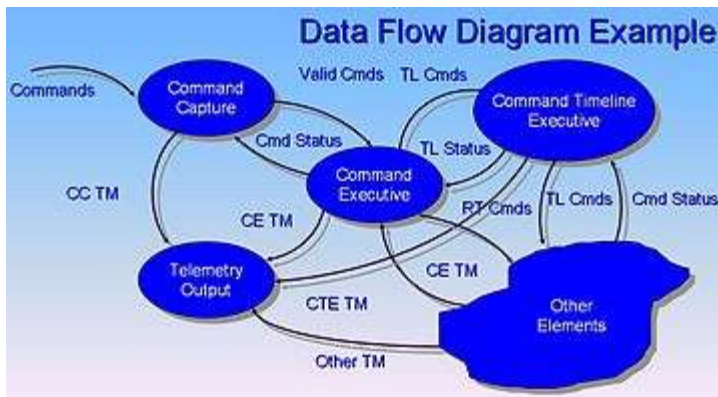
## 8. What is the objective of project planning?

**Ans:-**The aims and objectives will give your project a direction and will allow you to focus on results. Throughout the project, revisit your objectives and measure what you have achieved. At the end of the project they will help you to demonstrate what you have achieved. Following are the major objective of project planning:-

- **Specific** Be clear about what will be achieved
- **Measurable** Quantify results and measure when they have been achieved
- **Achievable** Ensure they *can* be achieved
- **Realistic** Can be attained with within project resources
- **Timed** Can be attained within a specified timescale.

## 9. What is DFD?

**Ans:-A Data Flow Diagram (DFD)** is a graphical representation of the "flow" of data through an information system, modeling its *process* aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).



## 10. What is prototype model?

**Ans:-The** basic idea here is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the currently known requirements. By using this prototype, the client can get an "actual feel" of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system. Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements. The prototypes are usually not complete systems and many of the details are not built in the prototype. The goal is to provide a system with overall functionality.

## 11. What is UML?

**Ans:-The Unified Modeling Language (UML)** is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system

## 12. What is estimation?

**Ans:-The Constructive Cost Model (COCOMO)** is an algorithmic software cost estimation model developed by Barry W. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current as well as future project characteristics. First published in Boehm's 1981 book *Software Engineering Economics* as a model for estimating effort, cost, and schedule for software projects.

## 13. How many software testing are there .Explain blackbox, white box, unit, integration, system testing's?

**Ans:** -There are many types of testing used to test software major of them are:-

- **White-box testing** (also known as clear box testing, glass box testing, and transparent box testing and structural testing) tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).
- **Black-box testing**  
Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation. The tester is only aware of what the software is supposed to do, not how it does it. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing and specification-based testing. One advantage of the black box technique is that no programming knowledge is required.
- **Unit testing**  
Unit testing, also known as component testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors.  
These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to assure that the building blocks the software uses work independently of each other.
- **Integration testing**  
Integration testing is any type of software testing that seeks to verify the interfaces between Components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be localized more quickly and fixed.  
Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.
- **System testing**  
System testing tests a completely integrated system to verify that it meets its requirements.\
- **System integration testing**  
System integration testing verifies that a system is integrated to any external or third-party systems defined in the system requirements.

#### 14. Define software metrics and software measurement?

**Ans:** - **Software measurement** is an indication of the size, quantity, amount or dimension of a particular attribute of a product or process. For example the *number of errors in a system* is a measurement.



**Software Metric** is a measurement of the degree that any attribute belongs to a system, product or process. For example the *number of errors per person hours* would be a metric.

Thus, software measurement gives rise to software metrics.

[Source](#)

### 15.Explain SDLC with diagram?

**Ans:-**SDLC stands for system development life cycle also referred to as the **application development life-cycle**, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system. The systems development life-cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.



### 16.Explain software reuse and reengineering?

**Ans:-**

**Software Reuse:** - In computer science and software engineering, **reusability** is the use of existing assets in some form within the software product development process. More than just code, assets are products and by-products of the software development life cycle and include software components, test suites, designs and documentation. Leverage is modifying existing assets as needed to meet specific system requirements. Because reuse implies the creation of a separately maintained version of the assets, it is preferred over leverage.



**Software reengineering:** -it is the process of discovering the technological principles of a human made device, object or system through analysis of its structure, function and operation. It often involves taking something (e.g., a mechanical device, electronic component, or software program) apart and analyzing its workings in detail to be used in maintenance, or to try to make a new device or program that does the same thing without using or simply duplicating (without understanding) any part of the original.

## 17.What is SQA?

**Ans: - Software quality assurance (SQA)** consists of a means of monitoring the software engineering processes and methods used to ensure quality. The methods by which this is accomplished are many and varied, and may include ensuring conformance to one or more standards, such as ISO 9000 or a model such as CMMI.

SQA encompasses the entire software development process, which includes processes such as requirements definition, software design, coding, source code control, code reviews, software configuration management, testing, release management, and product integration. SQA is organized into goals, commitments, abilities, activities, measurements, and verifications.



## Five to Eight Marks Type Question

### 1. What is system testing .Differentiate between black box and white box testing?

**Ans: - System testing** of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

S.N.	Black Box Testing	White Box Testing
1	The Internal Workings of an application are not required to be known	Tester has full knowledge of the Internal workings of the application
2	Also known as closed box testing, data driven testing and functional testing	Also known as clear box testing, structural testing or code based testing
3	Performed by end users and also by testers and developers	Normally done by testers and developers
4	Testing is based on external expectations - Internal behavior of the application is unknown	Internal workings are fully known and the tester can design test data accordingly
5	This is the least time consuming and exhaustive	The most exhaustive and time consuming type of testing
6	Not suited to algorithm testing	Suited for algorithm testing
7	This can only be done by trial and error method	Data domains and Internal boundaries can be better tested



Microsoft Office  
Excel Worksheet

### 2. What do you mean by SRS? Explain with functional and non functional requirement?

**Ans: -** A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform.

An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-world situations. Parameters such as operating speed, response time, availability, portability, maintainability, footprint, security and speed of recovery from adverse events are evaluated. Methods of defining an SRS are described by the IEEE (Institute of Electrical and Electronics Engineers) specification 830-1998.

**Non functional requirements are:-**

**Safety**

Information transmission should be securely transmitted to server without any changes in information

### **Reliability**

As the system provides the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.

### **Availability**

If the internet service gets disrupted while sending information to the server, the information can be sending again for verification.

### **Security**

The main security concern is for users account hence proper login mechanism should be used to avoid hacking. The tablet id registration is way to spam check for increasing the security. Hence, security is provided from unwanted use of recognition software.

### **Usability**

As the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

### **Functional requirements:-**

- Enable a user to view questions and their corresponding answers.
- Enable a logged in user to ask questions.
- Enable a logged in user to post answers.
- Enable a logged in user to up vote and down vote answers.
- Provide an interface for the admin to approve posts so that posts are not visible without admin approval
- Enable the admin to generate reports which contains all the posts and their corresponding replies

•

### 3. What is software process model what are different paradigm of software process model available in software engineering. Explain waterfall model.

**Ans: -Software process model:-**

A **software process model** is an abstract representation of a **process**. It presents a description of a **process** from some particular perspective. Separate and distinct phases of specification and development.

The software development strategy is referred as Software Engineering Paradigm. The software development strategy consists of methods, tools, and procedures. There exist various software development strategies or process models.

**Software Development Models:-**

- Water Fall Model.
- Iterative Waterfall Model.
- Prototyping Model.
- Evolutionary Model.
- RAD (Rapid Application Development) Model.
- Spiral Model.

Each model has its own specific steps for software development. A suitable development model is selected by considering several factors like requirement, application type, and application software to be used for development etc

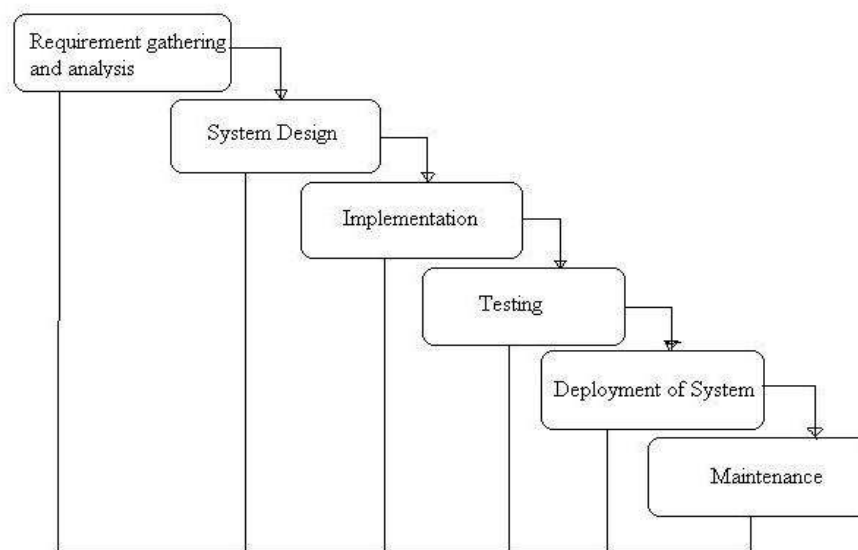
### 4. Which model is used to develop small application what are its advantages and disadvantages justify your answer.

**Ans: -**Waterfall model is used to develop small application.

The Waterfall Model was first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of model is basically used for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model the testing starts only after the development is complete. In **waterfall model phases** do not overlap.

**Diagram of Waterfall-model:**

### General Overview of "Waterfall Model"



#### Advantages of waterfall model:

- This model is simple and easy to understand and use.
- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- In this model phases are processed and completed one at a time. Phases do not overlap.
- Waterfall model works well for smaller projects where requirements are very well understood.

#### Disadvantages of waterfall model:

- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.

#### 5. Explain testing process and testing criteria.

**Ans: -Testing process:** - is a process rather than a single activity. This process starts from test planning then designing test cases, preparing for execution and evaluating status till the test closure. So, we can divide the activities within the fundamental test process into the following basic steps:

- 1) Planning and Control
- 2) Analysis and Design
- 3) Implementation and Execution
- 4) Evaluating exit criteria and Reporting
- 5) Test Closure activities

#### **Test criteria:-**

Software has to be evaluated and tested after every step during its development. Testing the program code is searching for errors. With large modular software programs the task of testing is often split into unit testing, integration testing, and system testing.

Test techniques developed in the past are mostly concerned with testing small programs, sub-systems or modules. Test data selection criteria and coverage measures, leading to statements about the completeness of the chosen tests, exist for those techniques.

The integration test of large modular software systems is more complicated. The number of existing test techniques suitable to reveal especially interface errors is smaller and the use of coverage measures is not very common.

Integration tests of real-world software systems (consisting of modules or components) are often carried out unsystematically. A common opinion is that more comprehensive integration tests are too costly. For economic reasons the test phase is often reduced to spot-checks (the well-known 'build-it-cheap-and-fix-it-later' approach).

The aim of this contribution is to introduce criteria and measures the use of which may lead to a systematic, reproducible process to integration testing of systems written in a procedural language. Test cases can be specified to satisfy the specified criteria and coverage measures can be used to check the completeness. The goal is to cover a high percentage of interface uses with a minimum of test cases.

## **6. Explain RMMM.**

**Ans:** -The RMMM Plan maybe developed in the form of a document. Alternatively, an organization may create a set of Risk Information Sheets (RIS) [often in electronic form] that contain all pertinent information outlined below. IMPORTANT note: Like most software engineering documents, the RMMM Plan evolves over time.

The goal of the risk mitigation, monitoring and management plan is to identify as many potential risks as possible. To help determine what the potential risks are and an effective strategy must consider three issues:

- **Risk avoidance.**
- **Risk monitoring.**
- **Risk management and contingency planning.**

If a software team adopts a proactive approach to risk, avoidance is always the best strategy. This is achieved by developing a plan for risk mitigation.

When all risks have been identified, they will then be evaluated to determine their probability of occurrence, and how Game Forge will be affected if they do occur. Plans will then be made to avoid each risk, to track each risk to determine if it is more or less likely to occur, and to plan for those risks should they occur.

It is the organization's responsibility to perform risk mitigation, monitoring, and management in order to produce a quality product. The quicker the risks can be identified and avoided, the smaller the chances of having to face that particular risk's consequence. The fewer consequences suffered as a result of good RMMM plan, the better the product and the smoother the development process.

## **7. What is software risk? Explain the different type of risk that occurs in software development?**

**Ans:** -Although there has been considerable debate about the proper definition for software risk, there is general agreement that risk always involves two characteristics. Although there has been considerable debate about the proper definition for software risk, there is general agreement that risk always involves two characteristics.

When risks are analyzed, it is important to quantify the level of uncertainty and the degree of loss associated with each risk. To accomplish this, different categories of risks are considered.

*Project risks* threaten the project plan. That is, if project risks become real, it is likely that project schedule will slip and that costs will increase. Project risks identify potential budgetary, schedule, personnel (staffing and organization), resource, customer, and requirements problems and their impact on a software project.

## **8. What is risk identification & risk projection?**

**Ans:** -The objectives of risk identification are to identify and categorize risks that could affect the project and document these risks. The outcome of risk identification is a list of risks. What is done with the list of risks depends on the nature of the risks and the project. On noncomplex, low-cost projects with little uncertainty (few risks); the risks may be kept simply as a list of red flag items. The items can then be assigned to individual team members to watch throughout the project development process and used for risk allocation purposes, as described later in this document. On complex, high-cost projects that are by nature uncertain, the risks can feed the rigorous process of assessment, analysis, mitigation and planning, allocation, and monitoring and updating described in this document.

**Risk projection:** - Also called risk estimation, attempts to rate each risk in two ways—the likelihood or probability that the risk is real and the consequences of the problems associated with the risk, should it occur. The project planner, along with other managers and technical staff, performs four risk projection activities: -

- (1) Establish a scale that reflects the perceived likelihood of a risk,
- (2) delineate the consequences of the risk,
- (3) Estimate the impact of the risk on the project and the product, and
- (4) Note the overall accuracy of the risk projection so that there will be no misunderstandings.

## **9. What is software reliability?**

**Ans:** -According to ANSI, Software Reliability is defined as: the probability of failure-free software operation for a specified period of time in a specified environment. Although Software Reliability is defined



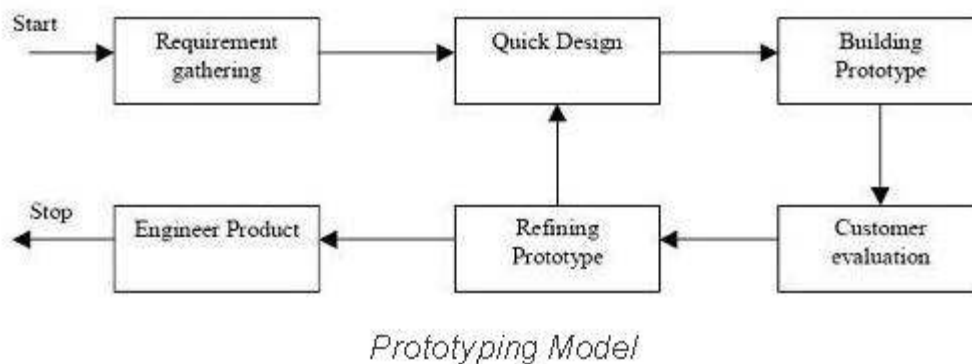
as a probabilistic function, and comes with the notion of time, we must note that, different from traditional Hardware Reliability, Software Reliability is not a direct function of time. Electronic and mechanical parts may become "old" and wear out with time and usage, but software will not rust or wear-out during its life cycle. Software will not change over time unless intentionally changed or upgraded.

Software failures may be due to errors, ambiguities, oversights or misinterpretation of the specification that the software is supposed to satisfy, carelessness or incompetence in writing code, inadequate testing, incorrect or unexpected usage of the software or other unforeseen problems.

## 10. What is software prototyping? Explain different types of prototyping available in software engineering?

**Ans:** -The basic idea here is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the currently known requirements. By using this prototype, the client can get an “actual feel” of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system. Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements. The prototypes are usually not complete systems and many of the details are not built in the prototype. The goal is to provide a system with overall functionality.

### Diagram of Prototype model:



### Advantages of Prototype model:

- Users are actively involved in the development
- Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed.
- Errors can be detected much earlier.

- Quicker user feedback is available leading to better solutions.
  - Missing functionality can be identified easily
  - Confusing or difficult functions can be identified
- Requirements validation, Quick implementation of, incomplete, but functional, application.

#### **Disadvantages of Prototype model:**

- Leads to implementing and then repairing way of building systems.
- Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.
- Incomplete application may cause application not to be used as the full system was designed Incomplete or inadequate problem analysis.

### **11.Explain object oriented software engineering.**

**Ans:** -Object Orientation is a software development methodology that is based on real world system. An object is the core concept involved in object orientation.

An object is the representation of a real world entity.

Object oriented view provides an abstraction that enables us to model the world in ways that help us to better understand and navigate. Therefore, software engineering follows the same concepts as conventional approaches

Analysis (identifies (object and classes (that are relevant to the problem domain)

Design -- (Identifies and provide (Architecture, Interfaces, component Level Detail

Implementation (using OO Language) (Transforms, Design (Code.

Testing (exercise (Object oriented architecture Interface & component.

### **12.What do you mean by software design process? What are the steps used in software design principle and design concepts.**

**Ans: - Software Design:** It is the first step in moving from problem domain to solution domain. The purpose of the design phase is to plan a solution of the problem specified by the requirements document. Starting with what is needed; design takes towards how to satisfy the needs.

### **13.What is the importance of process model in software engineering? Explain any three commonly process model?**

**Ans: -**

#### **14.Explain RAD? Advantages and Disadvantage.**

**Ans: -** RAD model is Rapid Application Development model. It is a type of incremental model. In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype. This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements.

**Diagram of RAD-Model:**

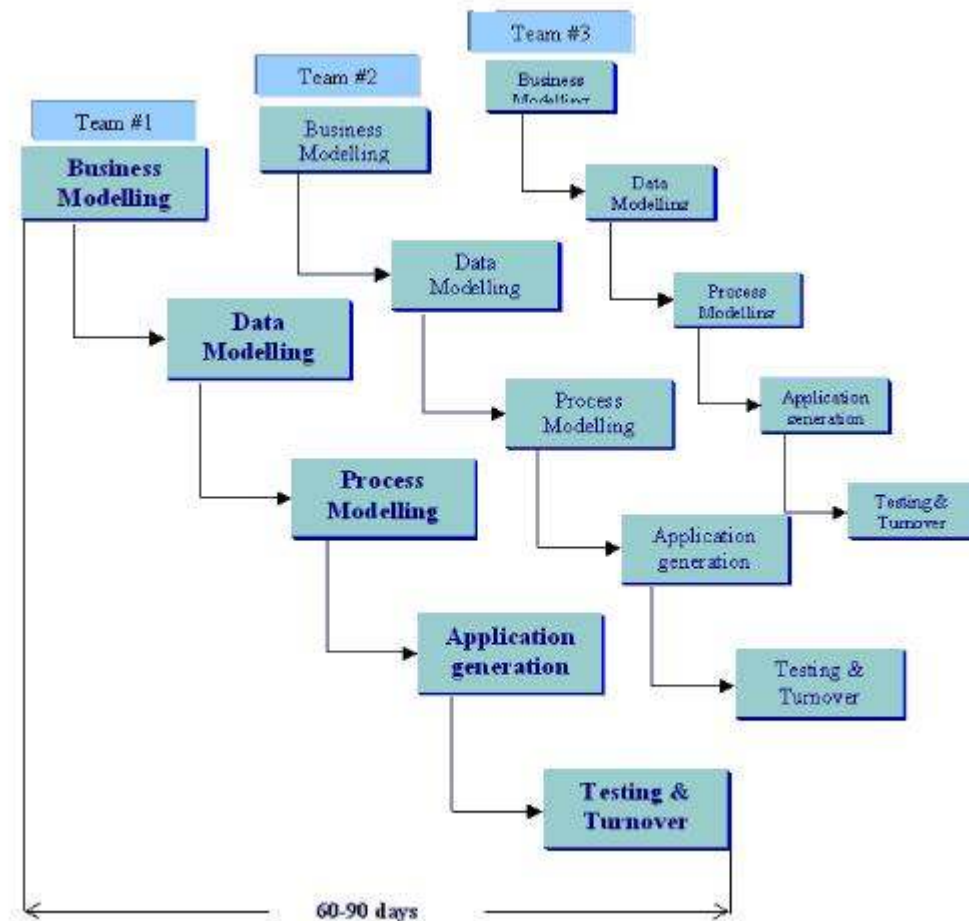


Figure 1.5 – RAD Model

The phases in the rapid application development (RAD) model are:

**Business modeling:** The information flow is identified between various business functions.

**Data modeling:** Information gathered from business modeling is used to define data objects that are needed for the business.

**Process modeling:** Data objects defined in data modeling are converted to achieve the business information flow to achieve some specific business objective. Description are identified and created for CRUD of data objects.

**Application generation:** Automated tools are used to convert process models into code and the actual system.

**Testing and turnover:** Test new components and all the interfaces.

#### **Advantages of the RAD model:**

- Reduced development time.
- Increases reusability of components
- Quick initial reviews occur
- Encourages customer feedback
- Integration from very beginning solves a lot of integration issues.

#### **Disadvantages of RAD model:**

- Depends on strong team and individual performances for identifying business requirements.
- Only system that can be modularized can be built using RAD
- Requires highly skilled developers/designers.
- High dependency on modeling skills
- Inapplicable to cheaper projects as cost of modeling and automated co degeneration is very high.

### **15.Explain BPR (business process reengineering)?**

**Ans:** - BPR seeks to help companies radically restructure their organizations by focusing on the ground-up design of their business processes. According to Davenport (1990) a business process is a set of logically related tasks performed to achieve a defined business outcome. Re-engineering emphasized a holistic focus on business objectives and how processes related to them, encouraging full-scale recreation of processes rather than iterative optimization of sub processes.

### **16.What is the analysis phase of software engineering considered? What approaches can be taken to insure it is successful?**

**Ans:** - The **analysis phase** defines the **requirements** of the system, independent of how these requirements will be accomplished. This phase defines the problem that the customer is trying to solve. The deliverable result at the end of this phase is a requirement document. Ideally, this document states in a clear and precise fashion what is to be built. This analysis represents the ``what" phase.

**Table 16.1:** The Analysis Phase: What does the system do?

Phase	Deliverable
Analysis	<ul style="list-style-type: none"> <li>● Requirements Document</li> </ul>
	<ul style="list-style-type: none"> <li>● Domain Ontology</li> </ul>
	- Things
	- Actions
	- States
	<ul style="list-style-type: none"> <li>● Typical Scenarios</li> </ul>
	<ul style="list-style-type: none"> <li>● Atypical Scenarios</li> </ul>

The requirement document tries to capture the requirements from the customer's perspective by defining goals and interactions at a level removed from the implementation details. The analysis phase is summarized in Table 16.1.

The **requirement** document may be expressed in a formal language based on mathematical logic. Traditionally, the requirement document is written in English or another written language.  
<http://infolab.stanford.edu/~burbach/watersluice/node4.html>

## 17. Define IEEE?

**Ans: -** The **Institute of Electrical and Electronics Engineers Standards Association (IEEE-SA)** is an organization within IEEE that develops global standards in a broad range of industries, including: power and energy, biomedical and health care, information technology, telecommunication, transportation, nanotechnology, information assurance, and many more.

IEEE-SA has developed standards for over a century, through a program that offers balance, openness, fair procedures, and consensus. Technical experts from all over the world participate in the development of IEEE standards.

## 18. Explain in brief about all the process model and their advantages and disadvantages?

**Ans: -**

## 19. What is software project planning .Explain project?

**Ans: -** Planning provides a road map for the software development process.

- Objective of Software Project
- Planning Software Scope
- Resources

- **Objective of Software Project planning :-**

The objective of software project planning is to provide a frame work that enables the manager to make reasonable estimates of resources, cost and schedule.

- **Software Scope**

The first activity in software project planning is the determination of software scope. A software project scope must be unambiguous and understandable at the management and technical levels. The software scope means the actual operation that is going too carried out by the software and its plus points and limitations.

- **Resource**

The second task of software planning is the estimation of resources required. Each resource is specified with the following characteristic. Resource descriptions, details of availability, when it is required, how long it is required. Human Resource Hardware Resource Software Resource People are the primary software development resource. The planners evaluate the scope and select the appropriate people for appropriate positions.

## **20.Explain software project scheduling .what is the reason for late software delivery?**

**Ans: -**

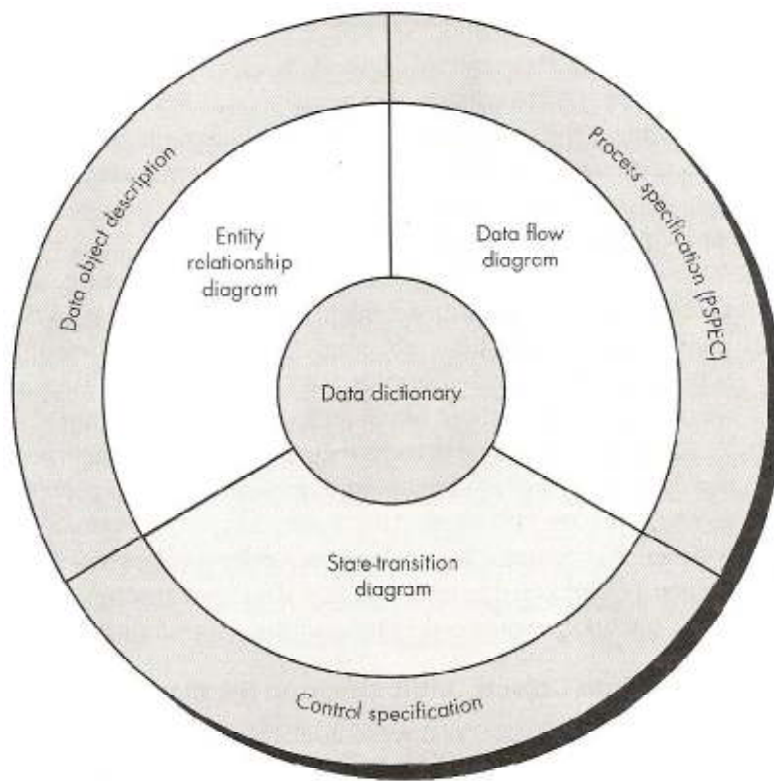
## **21.Explain different element of Analysis modeling?**

**Ans: -** **THE ELEMENTS ANALYSIS MODEL**

The analysis model must achieve three primary objectives:

- (1) To describe what the customer requires.
- (2) To establish a basis for the creation of a software design.
- (3) To define a set of requirements that can be validated once the software is built.

To accomplish these objectives, the analysis model derived during structured analysis takes the form illustrated in



**Figure:-21.1 the structure of the analysis model**

At the core of the model lies the data dictionary → a repository that contains descriptions of all data objects consumed or produced by the software. Three different diagrams surround the core. The entity relation diagram (ERD) depicts relationships between data objects. The ERD is the notation that is used to conduct the data modeling activity. The attributes of each data object noted in the ERD can be described using a data object description. The data flow diagram (DFD) serves two purposes:

- (1) To provide an indication of how data are transformed as they move through the system and
- (2) To depict the functions (and sub-functions) that transforms the data flow.

The DFD provides additional information that is used during the analysis of the information domain and serves as a basis for the modeling of function. A description of each function presented in the DFD is contained in a process specification (PSPEC).



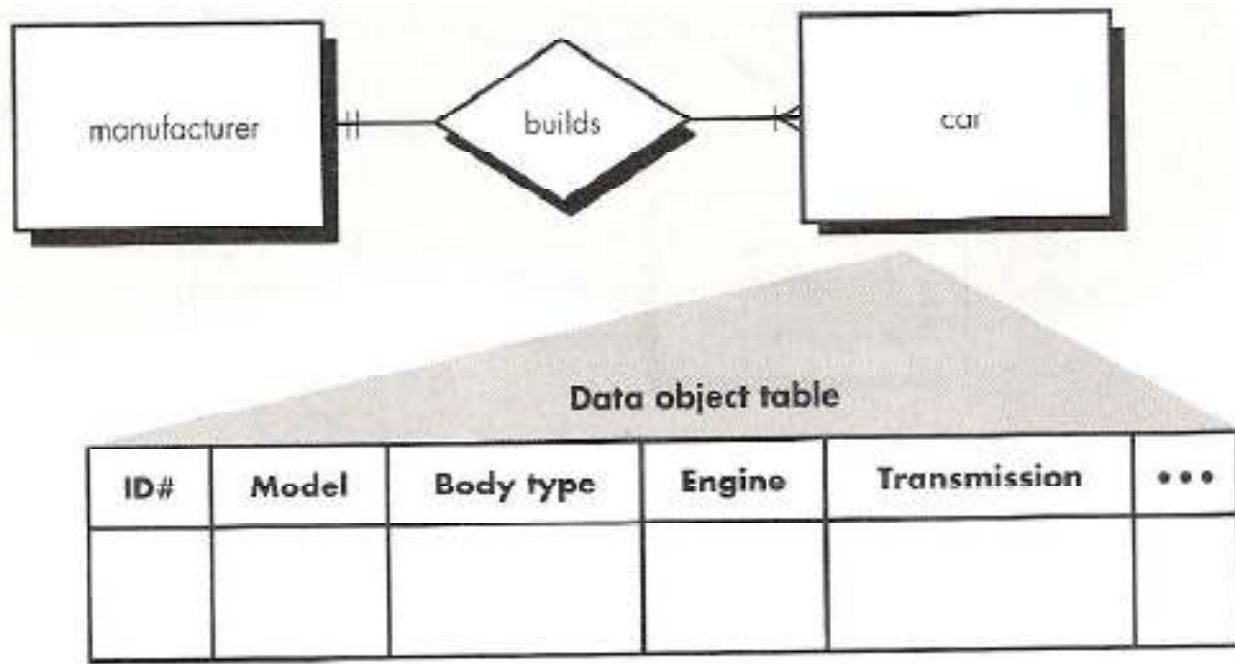


Figure:-21.2 Data object table

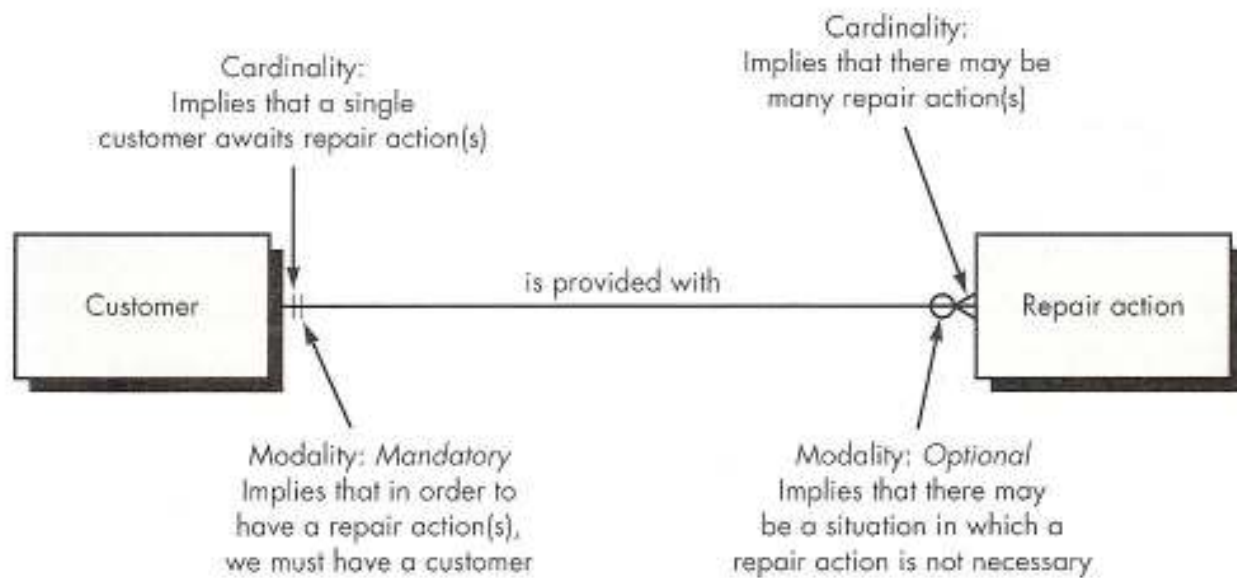


Figure:-21.3 Cardinality



(a) A basic connection between objects

**Figure:-21.4 Relationship**